OBJECT CLASSIFICATION

Field of the Invention

This invention generally relates to classification of objects and, more specifically, to classification of objects using optimal combinations of underlying features.

Background of the Invention

Modern society creates a sea of data. It can be difficult to understand large data sets using standard data analysis tools. The problem is particularly acute for data sets containing many objects, with many measured properties for each object. The typical approaches of plotting one parameter against another, computing histograms, measuring correlations, and so on are simply insufficient for exploring the data when there are more than a handful of parameters for each object in the sample. Object classification is a very useful tool for data exploration in large, complex problems as it can provide an accurate, understandable characterization of a complex data set.

A classifier takes a set of parameters (or features) that characterize objects (or instances) and uses them to determine the type (or class) of each object. The classic example in astronomy is distinguishing stars from galaxies. For each object, one measures a number of properties (brightness, size, ellipticity, etc.); the classifier then uses

these properties to determine whether each object is a star or a galaxy. Classification of objects on the basis of their possession of a diversity of features, however, is a problem of widespread application. Classifiers need not give simple yes/no answers -- they can also give an estimate of the probability that an object belongs to each of the candidate classes.

The classification process generally comprises three broad steps. The first is adjustment of input data, the second is classification, and the third is cross-validation. The first step mainly involves noise reduction and/or normalization and is highly domain dependent; it is essential for a proper interpretation of the results. The second step is domain independent. In the third step, the accuracy of the classifier is measured. Knowledge of the accuracy is necessary both in the application of the classifier and also in comparison of different classifiers. Accuracy is typically determined by applying the classifier to an independent training set of objects with known classifications. The advantage of cross-validation is that all objects in the training set get used both as test objects and as training objects. Often steps two and three are carried out repeatedly until a satisfactory classifier has been obtained. Various standard cross validation techniques are known, one of which is five-fold cross-validation.

Generally the computationally hard part of classification is inducing a classifier, i.e., determining the optimal (or at least good) values of whatever parameters the

classifier will use. The classification problem becomes very hard when there are many parameters. There are so many different combinations of parameters that techniques based on exhaustive searches of the parameter space are computationally infeasible. Practical methods for classification always involve a heuristic approach intended to find a ``good-enough" solution to the optimization problem. There are numerous recognized approaches in the art, including neural networks, neural-neighbor classifiers, axis parallel decision trees, and oblique decision trees. Such approaches are discussed, for example, in Aho, Hopcroft, Ulman "The Design and Analysis of Computer Algorithms", Addison Wesley Publishing Co, 1976. There is, however, need for improvement so as have more accurate classifications.

Summary of the Invention

In accordance with at least one presently preferred embodiment of the present invention, there is broadly contemplated a system and method classification of objects using optimal combinations of underlying features.

In summary, one aspect of the invention provides a system for classifying objects, the system comprising: an arrangement for formulating a query to identify objects having properties of interest; an arrangement for selecting properties of the objects to compare

with object properties included in the query; and an arrangement for determining if based on the selected properties if object belongs in the query.

Another aspect of the present invention provides a method for classifying objects, the method comprising the steps of: identifying properties of objects; formulating a query to identify objects having properties of interest; selecting properties of the objects to compare with object properties included in the query; and determining if based on the selected properties if the object belongs in the query.

Furthermore, an additional aspect of the invention provides a program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for classifying objects, said method comprising the steps of: identifying properties of objects; formulating a query to identify objects having properties of interest; selecting properties of the objects to compare with object properties included in the query; and determining if based on the selected properties if the object belongs in the query.

For a better understanding of the present invention, together with other and further features and advantages thereof, reference is made to the following description, taken in conjunction with the accompanying drawings, and the scope of the invention will be pointed out in the appended claims.

Brief Description of the Drawings

Fig. 1 is a flow chart depicting overview of the method of the present invention;

Fig. 2 shows the universal set, query set, and sets; and

Fig. 3 shows the query set and the result sets in accordance with the invention

Description of the Preferred Embodiments

The present invention provides an improved system and method for classifying objects on the basis of their possession of a diversity of features using the best boolean expression that represents the most optimal combination of the underlying features.

Referring now to Figure 1, a flow chart is depicted which generally shows an overview of the method of the present invention. At Step S10, properties of items are identified. At Step S20, the query function is determined, that is, those properties an item returned should possess. At Step S30, those items which best define the query are determined. At Step S40, it is determined whether the item belongs in the query based upon the selected properties. Steps S20 and S30 are generally considered to be the classifer and Step S40 is generally considered to be cross-validation.

Referring now to Figure 2, U is the universal set of all objects, where objects may display zero, one or more features. The collection of objects from U that posses a specific feature i is referred to as the object set Si with $1 \le i \le n$. Q represents a defined collection of objects from U that are known to below to a particular class or classification bucket. For a given $k \le n$, the objective is to obtain the best combination of some k features that constitutes the most specific and sensitive signature characteristic of the object collection Q. Usually membership to Q is a non-obvious attribute. For instance, Si(s) are a priori attributes of an event and Q is the a posteriori outcome.

Where $U = \{e_1, e_2, \ldots, e_{n'}\}$ is the universe of n' elements, $S_j \subseteq U$, $1 \le j \le n$ and $Q \subseteq U$ and $k \le n$, the task is to find an *expression on any k sets* S_j that *best defines* Q. Each set S_j corresponds to a feature A_j defined to have some value v_{jl}. There are two issues that need further clarification here: first, what is meant by "expression on the sets", and second, the defintion of "best defines" Q. The former gives the form of the output and the latter defines the optimizing function.

Form of the Output: Given a set S that has the value of v of feature A, an expression can be written of the form $S(A=v)$. Just as sets can be combined using union (OR), intersection (AND), complement (NOT) operations so can the corresponding

expressions using the $\vee$, $\wedge$, and ~ operations respectively. Let A^1 and A^2 denote two expressions. Then recall the following:

1. $S(A^1 \vee A^2) = S(A^1) \cup S(A^2)$

2. $S(A^1 \wedge A^2) = S(A^1) \cap S(A^2)$

3. $S(\sim A^1) = \tilde{S}(A^1)$

4. $S_1 - S_2 = S_1 \wedge \tilde{S}_2$

Thus given any logical expression *LE* on the expressions, there exists a set corresponding to it given by *S*(*LE*).

Optimizing Function: An expression needs to be found that "best" describes *Q*. To define "best", the following definitions are used. The false and true positives and, the false and true negatives have the usual meaning: however, to establish the notation we define them below:

Definition 1 (false positives, false negatives) Given a set $S \subseteq U$ and a set $Q \subseteq U$, the false positives of *S* with respect to *Q*, given as *FP*(*S*,*Q*) is defined to be

$$FP(S,Q) = \{x \mid x \in S, x \notin Q\}$$

and the false negatives of S with respect to Q, given as $FN(S,Q)$ is defined to be

$$FN(S,Q) = \{x \mid x \notin S, x \in Q\}$$

Definition 2 (true positives, true negatives) Given a set $S \subseteq U$ and a set $Q \subseteq U$, the true positives of S with respect to Q, given as $TP(S,Q)$ is defined to be

$$TP(S,Q) = \{x \mid x \in S, x \in Q\}$$

and the true negatives of S with respect to Q, given as $TN(S,Q)$ is defined to be

$$TN(S,Q) = \{x \mid x \notin S, x \notin Q\}$$

The errors are the false positives and the false negatives. The "best" fit reduces these errors. However, false positives and false negatives need not have the same relative weight and two constants $\alpha, \beta \geq 0$ are introduced to deal with this. The problem may be formally stated as over all the first order logical expressions or boolean expressions on the features, find the one that optimizes the following function"

$$\min_{LE} \{\alpha \mid FP(S(LE),Q) \mid + \beta \mid FN(S(LE),Q) \mid\}$$

$\alpha,\beta>0$.

Attributes: Depending on how two values v_1 and v_2 of an attribute A can be compared, there may be two kinds of attributes. Let $v_1 \equiv v_2$ imply that the two attribute values are equivalent in the application.

1. non-numeric: $v_1 \equiv v_2 \Leftrightarrow v_1 = v_2$.

2. numeric: $v_1 \equiv v_2 \Leftrightarrow | v_1 - v_2 | \leq \delta$ for some given $\delta \geq 0$.

Numeric attributes give the option of exploiting the complete ordering that exists between the values. Hence an expression with integral v, such as $((v=3) \vee (v=4) \vee (v=5))$ can be written simply as $(3 \leq v \leq 5)$. Also, an expression with real values of v, such as $((v > 2.5) \wedge (v < 3.2))$ can be written as $(2.5<v<3.2)$.

The disclosure will now turn to a discussion of the algorithm used in the present invention. The algorithm works in the following two steps:

1. The problem is solved exactly for the $k=n$ case (see below). Obtain the optimal cost O.

2. From this optimal configuration, using a greedy algorithm one attribute at a time is dropped to obtain the given k attributes.

(a) Dummy points $x_j \in U$, $1 \leq j \leq n'$ are introduced such that $x_j \in S_i$, $1 \leq i \leq n$ and $x_j \in Q$, $1 \leq j \leq n'$.

(b) At each iteration the attribute S_i that increases the cost by the minimum amount is removed.

The disclosure will now turn to a discussion of solving the $k=n$ case. The algorithm to find the boolean expression (over *all* possible boolean expressions) of sets proceeds in the two steps set forth below:

1. Partition the union of the S sets U' into K non-overlapping sets S'_k such that $S'_i \cap S'_j = \phi$, for all $i \neq j$. Each S'_k corresponds to a boolean expression on the given sets S_i. Clearly, $K \leq n'$.

2. Given the K sets S'_k, we wish to obtain an expression on these that minimizes the error. Define the following sets: $C_k = Q \cap S'_k$ and $D_k = S'_k - C_k$. By our defintions $C_k = TP(S'_k, Q)$ and $D_k = TN(S'_k, Q)$. Let $B = Q - \bigcup_{k \in K} S'_k$. See Figure 3 for a pictorial depiction of these sets. In this figure Q is the query set and the S' sets are the mutually non-overlapping sets obtained in Step 1. The true positives of each S' is shown as C and the true negatives as D.

$$S_i' \cap S_j' = \phi, \forall i \neq j \quad (1)$$

$$\overline{S_i'} = U' - S_i = \bigcup_{j \neq i} S_j' \quad (2)$$

Using Equation 2, the final expression can have only S_i' terms (and not $\overline{S_i'}$) or vice versa. And using Equation 1, the only meaningful operation is union (and not intersection) of the S_i'-terms or vice-versa.

Consider an arbitrary expression (union of S_i' terms): $S_{k_1}' \cup S_{k_2}' \cup \ldots S_{k_1}' = S$. Then clearly,

$$\mathrm{FP(S,Q)} = \bigcup_{i \in k_1 \ldots k_l} \mathrm{D_i}$$

and

$$FN(S,Q) = B \bigcup_{i \in k_1 \ldots k_l} C_i$$

Hence, in this step the optimal value can be obtained simply by traversing through S_i''s and making a local decision which guarantees a global minimum. The algorithm works as follows: Let *LE* be the boolean expression and *O*, the cost being minimzed.

$LE \leftarrow \phi$, $O \leftarrow 0$;
For $i=1...K$
Begin
if $\alpha|D_i| \leq \beta|C_i|$, then $LE \leftarrow LE \vee S_i'$, $O \leftarrow O+\alpha|D_i|$
else $O \leftarrow O+\beta|C_i|$
End
$O \leftarrow O+|B|$

As discussed above, classification of objects on the basis of their possession of a diversity of features is a problem of widespread application. The present invention may be applied to any number of applications, non-limiting examples of which are set forth in the Appendix hereto. The disclosure will now turn to a discussion of the present invention in a particular application.

The universal set is insurance policies (1 to 11) having three attributes, 1) the policy was made in year 2001 or later, 2) the make of the insured automobile is Toyota, and 3) the insured driver is under the age of 20. The 11 data points have the following attributes:

1) year 2001, not Toyota, driver age above 20
2) year 2001, not Toyota, driver age above 20
3) year 2001, not Toyota, driver age above 20
4) year 2001, Toyota, driver age above 20
5) year 2001, Toyota, driver age above 20
6) year 2001, Toyota, driver age above 20
7) year before 2001, Toyota, driver age above 20
8) year before 2001, Toyota, driver age above 20
9) year before 2001, Toyota, driver age 20 or below

10) year before 2001, Toyota, driver age 20 or below
11) year before 2001, Toyota, driver age 20 or below

The set of insurance claims, Q, is Q = {2, 3, 5, 6, 8, 10, 11}. Referring now to Figure 3:

S'1 = {1,2,3}	year 2001, not Toyota
S'2 = {4,5,6}	year 2001, Toyota
S'3 = {7,8}	Toyota, year before 2001, age above 20
S'4 = {9,10,11}	age 20 or below

C1 = {2,3}	D1= {1}
C2 = {5,6}	D2 = {4}
C3 = {8}	D3 = {7}
C4={10,11}	D4 = {9}

Assuming $\alpha = 1$ and $\beta = 1$, the Logical Expression is S'1 v S`2 v S`3 v S'4 with minimum cost 4.

It is to be understood that the present invention, in accordance with at least one presently preferred embodiment, has elements which may be implemented on at least one general-purpose computer running suitable software programs. These elements may also be implemented on at least one Integrated Circuit or part of at least one Integrated Circuit. Thus, it is to be understood that the invention may be implemented in hardware, software, or a combination of both.

If not otherwise stated herein, it is to be assumed that all patents, patent applications, patent publications and other publications (including web-based publications) mentioned and cited herein are hereby fully incorporated by reference herein as if set forth in their entirety herein.

Although illustrative embodiments of the present invention have been described herein with reference to the accompanying drawings, it is to be understood that the invention is not limited to those precise embodiments, and that various other changes and modifications may be affected therein by one skilled in the art without departing from the scope or spirit of the invention.